Dimension Embeddings for Monocular 3D Object Detection

Yunpeng Zhang[1,2], Wenzhao Zheng[1,2], Zheng Zhu[3], Guan Huang[3],
Dalong Du[3], Jie Zhou[1,2], Jiwen Lu[1,2*]
[1]Beijing National Research Center for Information Science and Technology, China
[2]Department of Automation, Tsinghua University, China
[3]PhiGent Robotics
{zhang-yp19, zhengwz18}@mails.tsinghua.edu.cn; zhengzhu@ieee.org;
{guan.huang, dalong.du}@phigent.ai; {jzhou, lujiwen}@tsinghua.edu.cn

Abstract

Most existing deep learning-based approaches for monocular 3D object detection directly regress the dimensions of objects and overlook their importance in solving the ill-posed problem. In this paper, we propose a general method to learn appropriate embeddings for dimension estimation in monocular 3D object detection. Specifically, we consider two intuitive clues in learning the dimension-aware embeddings with deep neural networks. First, we constrain the pair-wise distance on the embedding space to reflect the similarity of corresponding dimensions so that the model can take advantage of inter-object information to learn more discriminative embeddings for dimension estimation. Second, we propose to learn representative shape templates on the dimension-aware embedding space. Through the attention mechanism, each object can interact with the learnable templates and obtain the attentive dimensions as the initial estimation, which is further refined by the combined features from both the object and the attentive templates. Experimental results on the well-established KITTI dataset demonstrate the proposed method of dimension embeddings can bring consistent improvements with negligible computation cost overhead. We achieve new state-of-the-art performance on the KITTI 3D object detection benchmark.

1. Introduction

3D object detection aims to estimate the 3D locations, poses, and sizes of surrounding objects and serves as one fundamental task in sensing the 3D environment. It has been widely used in fields including autonomous driving and robot navigation. Most existing methods rely on point clouds from LiDAR devices [18, 40, 41, 51] or binocular images from stereo cameras [7, 20, 37, 45] for accurate 3D object detection. Though these methods can benefit from direct distance information and achieve impressive performance, the LiDAR sensors are still expensive for large-scale applications and stereo cameras can suffer from inaccurate calibrations. In comparison, monocular methods, which only utilize one color camera and reason about appearance information, have been a promising alternative and receiving increasingly more attention [1, 6, 8, 16, 23, 27, 32, 36, 50, 52, 55].

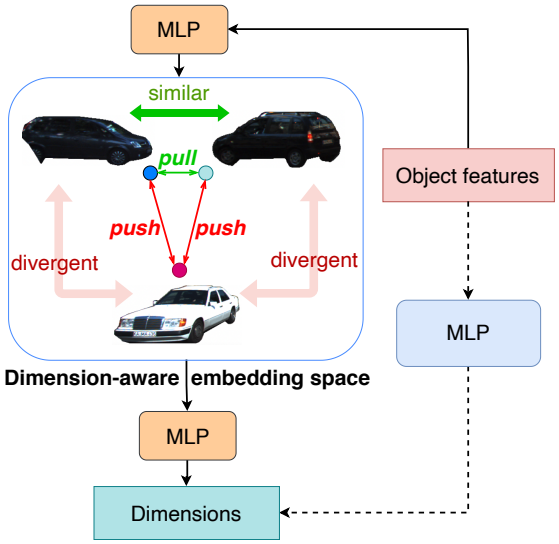

Figure 1. **Overview of the proposed dimension embeddings for monocular 3D object detection.** We first project the features of objects onto the dimension-aware embedding space, where objects with similar dimensions are pulled together and objects with divergent dimensions are pushed away. Compared with the direct method in the dotted line, our method based on dimension embeddings can utilize inter-object similarities as useful information to learn more discriminative features for dimension estimation.

Considering the ill-posed nature of monocular 3D object detection, the 3D dimension and the distance of objects are fundamentally entangled. Existing methods [3, 21, 26–28, 31,

*Corresponding author.

52, 55] usually regard the estimation of object distance as the key challenge and design various approaches to solve the problem. Some methods [26, 36, 52, 55, 59] directly regress the object depth with keypoints-based detectors. Recently, more methods [19, 21, 27, 28, 32, 55] compute the distance by combining the camera matrix, 2D geometric constraints, and 3D priors like the estimated dimensions and regressed 3D keypoints. The utilized 2D geometric constraints can include the sizes of 2D bounding boxes [19, 28] and predefined keypoints [21, 27, 55]. These more explainable methods can generate more accurate 3D locations and also generalize to different camera calibrations. To this end, we claim that the estimation of 3D dimensions is of vital importance for monocular 3D object detection. On the one hand, the 3D dimension directly serves as an element to form the desired 3D bounding box. On the other hand, solving the object distance with geometry heavily relies on the accuracy of 3D dimensions, because convolutional networks [11, 22, 54, 59] are skilled in predicting 2D bounding boxes and keypoints as accurate geometric constraints.

Though the importance of dimension estimation is clarified, most existing methods for monocular 3D object detection simply regress the absolute 3D dimension [27, 31, 52] or the relative offset with respect to the mean object dimension of each category [28, 55, 59]. To further improve the dimension estimation, we propose to leverage two important clues: (1) Since the consideration of similarity is fundamental for humans to reason about the world, we argue that modeling the similarities among dimensions of different objects is an effective way to provide additional information for dimension reasoning. (2) The mapping relationship from visual appearances to object dimensions is strongly related to the underlying subcategories. For example, the dimension of one car can be easily predicted if its specific type can be determined and has been memorized by the network.

To explicitly embed these clues into the neural network, we first propose to learn a dimension-aware embedding space where two objects are projected to be close to each other if they have similar dimensions and far apart otherwise, as illustrated in Figure 1. This is achieved by encouraging the pair-wise embedding distance to be proportional to the distance between their corresponding dimensions. In this way, the network can fully exploit the rich information of similarities among different objects and learn more discriminative features for dimension estimation. To further consider the information of latent subcategories, we define learnable shape templates on the embedding space and enable the interaction between extracted objects and the templates through the attention mechanism [46]. The attention scores and the template dimensions can be combined to produce the initial dimension estimations, which are further refined with the attentive features for final prediction.

We conduct extensive experiments on KITTI 3D object detection benchmark [10] with multiple state-of-the-art methods. The experimental results demonstrate our proposed dimension embedding is generalized, effective, and lightweight. The main contributions of the paper can be summarized in the following three aspects: (1) We propose to utilize the inter-object dimension similarities as useful information for dimension learning and formulate the solution as learning a dimension-aware embedding space. (2) We design the learnable shape templates to implicitly utilize the information of underlying subcategories. Through the attention mechanism, dimension embeddings of objects can aggregate attentive features and dimensions from these templates and formulate the dimension estimation as a coarse-to-fine refinement. (3) Our proposed dimension embedding module can be generally applied to existing keypoints-based methods for monocular 3D object detection and bring consistent improvement. We also achieve new state-of-the-art performance on the *test* set of KITTI 3D object detection benchmark, with real-time latency.

2. Related Work

2.1. Monocular 3D Object Detection

Tremendous progress has been made for monocular 3D object detection with many proposed approaches [1, 2, 6, 8, 16, 21, 23, 26–28, 31, 32, 42, 43, 52, 55, 59]. Considering the additional information utilized, these methods can be mainly categorized into depth-based and image-based methods. In depth-based methods, pseudo-LiDAR [50] first proposes to lift the estimated depth map into 3D points, which are further processed by LiDAR-based detectors [18, 35, 41] for detection. AM3D [30] proposes a multi-modal fusion module to enhance the pseudo-LiDAR features with clues from color images. PatchNet [29] organizes pseudo-LiDAR as the image representation and utilizes powerful 2D CNN to boost the detection performance. Though depth-based methods usually achieve better performance, they require dense depth maps for supervision and perform worse when applied to new scenes. In contrast, image-based methods do not rely on extra information for training. Mono3D [6] first samples candidates based on the ground prior and scores them with semantic/instance segmentation, contextual information, object shape, and location prior. Deep3DBox [32] proposes the *MultiBin* loss for orientation estimation and solves 3D bounding boxes with geometrical constraints from estimated 2D boxes. FQNet [23] trains an IoU prediction network with simulated data, which can accurately measure the localization accuracy of 3D candidates. MonoDIS [42] presents a disentangling transformation for the 3D detection losses to isolate different groups of parameters. MonoPair [52] utilizes the pair-wise relationships between neighboring objects to post-optimize the locations of objects. MonoFlex [55] designs the decoupled representation for truncated objects and

formulates the depth estimation as the uncertainty-guided ensemble learning.

Compared with existing methods, we focus on the overlooked problem of dimension estimation and is therefore orthogonal to other studies. MonoPair [52] considers the relationships between nearby objects, while our proposed dimension-aware embedding can utilize the similarity between every pair of objects within a batch of images.

2.2. Deep Metric Learning

Deep metric learning methods usually use convolutional neural networks to obtain an embedding for each image and employ the Euclidean distance between embeddings to measure the distance. They impose a discriminative loss on the embeddings to enlarge interclass distances and reduce intraclass distances [4, 9, 12, 33, 39, 44, 47–49, 53]. For example, the widely used triplet loss [39] enlarges the distance between the anchor and the negative sample as well as reduces the distance between the anchor and the positive sample to form a distance margin. The triplet loss suffers from the lack of informative triplets during training and motivates Movshovitz *et al.* [33] to assign a proxy for each class and instead impose constraints on the relations between samples and proxies.

All the aforementioned methods can only deal with data within discrete labels, and cannot handle structurally labeled data. Recently, Kim *et al.* [13] proposed a log-ratio loss to learn an embedding space with continuous labels. Zheng *et al.* [57] further presented a structural deep metric learning method to apply deep metric learning to the structural estimation problem. However, all these metric learning methods are exploited to model the similarities among entire images. In this paper, we further apply the similarity constraint on the level of extracted objects from a batch of images and propose the variance-normalized distance for dimensions. Also, most metric learning methods aim to solve the problem of retrieval while we utilize the information of similarities to improve the dimension estimation.

3. Approach

3.1. Problem Statement

With one single color image and the camera parameters as input, the target of monocular 3D object detection is to recognize and localize objects of interest precisely in the 3D space. For each concerned object, we need to predict its category and fit the object with a tight 3D bounding box, which is generally represented with the 3D location (x, y, z), the 3D dimension (h, w, l) and the orientation θ.

3.2. Baseline Model

Following the recent trend in monocular 3D object detection [21, 27, 28, 31, 52, 55, 59], we build our baseline model based on the fully convolutional detector Monodle [31] due to its simplicity and scalability. As shown in Figure 2, the framework employs the CNN backbone DLA-34 [54] to extract visual features and then implements multiple parallel heads for center classification, 2D detection, and 3D detection. The fundamental settings for each task are introduced in the following paragraphs.

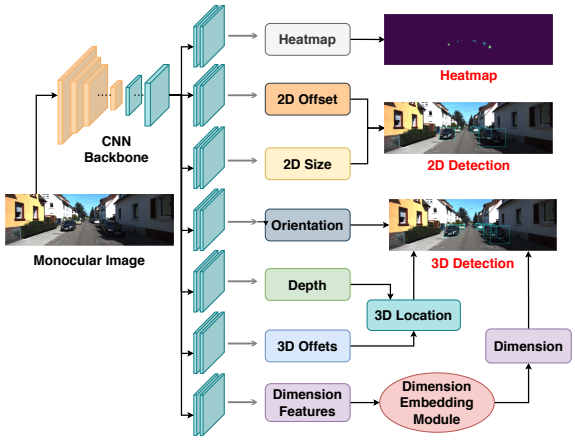

Figure 2. **Illustration of the keypoints-based 3D object detection framework.** With the feature map extracted from the input image by the CNN backbone, multiple parallel heads are attached to the shared feature map. The heatmap branch predicts the possibility that one location is the center of a potential object. The predictions from the branches of 2D offset and 2D size can recover the 2D bounding boxes, while the orientation, object depth, 3D offset, and dimension branches are collaborated to predict the 3D bounding boxes. Based on the general framework, we aim to improve the dimension estimation by considering inter-object similarities in the embedding space and learning representative shape templates.

Center-based classification. Following CenterNet [59], the classification branch identifies objects by determining whether each pixel on the feature map is the representative center of a potential object of interest. The classification prediction is formulated as a class-wise center heatmap with the size of $(H \times W \times c)$, where c refers to the number of interested categories. Following the design of Monodle [31], the projected 3D center instead of the 2D center is chosen as the representative center because the projected 3D center serves as one key factor in recovering the 3D location and can help to learn geometry-aware features. The corresponding loss $\mathcal{L}_{heatmap}$ follows the formulation of focal loss [22].

2D detection. To perform 2D detection, the offset between the projected 3D center and the 2D center and the size of the 2D bounding box are regressed. Though the 2D detection branch is not mandatory for 3D object detection, this simpler task can serve as the auxiliary supervision to make the features focus on interested objects [31, 55] and improve the performance. The loss functions for regressing 2D bounding box information are denoted as $\mathcal{L}_{offset2d}$ and $\mathcal{L}_{size2d}$.

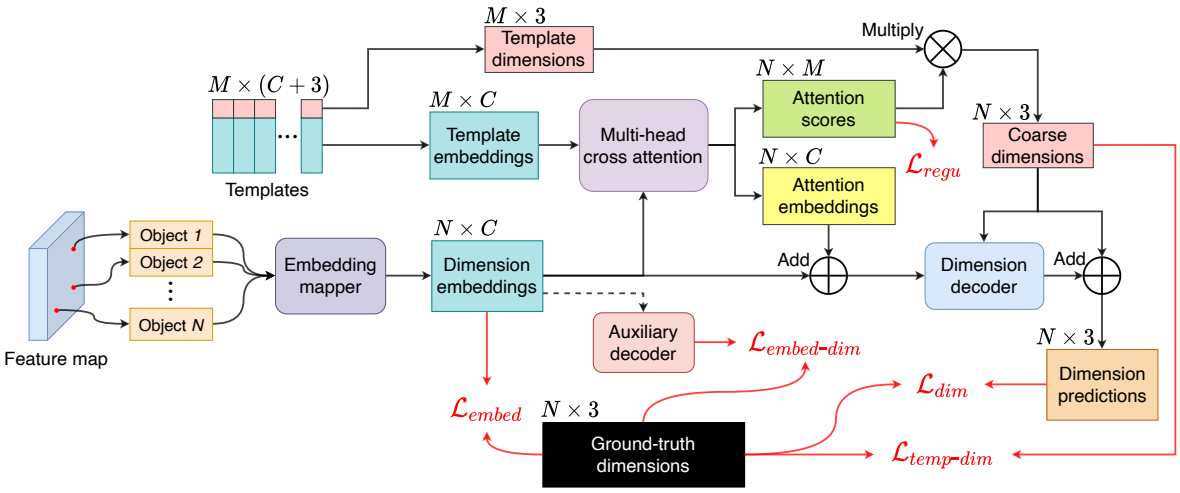

Figure 3. **Framework of the proposed shape embedding module.** Given the extracted feature map from the backbone CNN, we can obtain the features of N recognized objects by indexing their corresponding positions. To estimate the dimensions of these objects, their features are first mapped to the dimension-aware embedding space, where the dense log-ratio loss is imposed to encourage the embedding distance to reflect the dimension difference. Through the multi-head cross-attention module, these embeddings can then interact with the learnable shape templates, which include both template embeddings and template dimensions. Since the attention scores can reflect the similarities between objects and templates, we multiply the normalized scores with the template dimensions to obtain the coarse dimensions. To further refine the prediction, the object embeddings, the output attentive embeddings, and the coarse dimensions are combined and processed by the dimension decoder to predict the residual dimensions. (Best viewed in color.)

3D detection. The estimation of the 3D bounding box can decompose into the projected 3D center, depth, orientation, and dimension. Though the projected 3D center, as the selected representative center, can be predicted from the heatmap, an additional 3D offset branch is still required to compensate for the quantization error due to downsampling. The depth branch predicts the inverse depth and jointly models the uncertainty [31, 52]. For the orientation estimation, the local orientation is regressed with MultiBin loss [32]. For the 3D dimension estimation, the IoU oriented optimization [31] is utilized as the baseline and the proposed dimension embedding module is applied for our model. Therefore, the overall loss functions for 3D detection include $\mathcal{L}_{offset3d}$, $\mathcal{L}_{depth}$, $\mathcal{L}_{orien}$, and $\mathcal{L}_{dim}$.

3.3. Dimension Embedding Module

Formally, let $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N]$ be the features of N extracted objects from a batch of input images and $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, \cdots, \mathbf{d}_N]$ be the corresponding target dimensions, the dimension embedding module is proposed to learn the mapping function while also considering the dimension relationships among objects as useful information. As illustrated in Figure 3, the overall process can be divided into the following three steps. First, the embedding mapper takes the extracted object feature $\mathbf{x}$ and maps it to the dimension embedding $\mathbf{y}$, where the embedding distance is guided to reflect the inter-object relations of dimensions. Second, the attention mechanism is employed to correlate the dimension embeddings $\mathbf{y}$ and the learnable shape templates $\mathbf{T}$. With the computed attention scores, the coarse dimensions are formulated as the weighted averages of all template dimensions. Then, the dimension embeddings of objects and the attentive features from templates are combined to further refine the coarse dimension predictions. In Sec. 3.4, we introduce how the dimension-aware embedding space is learned to embed the information of dimensional similarities. In Sec. 3.5, we elaborate on the design of shape templates, how the attention mechanism is applied, and the residual refinement in dimension prediction.

3.4. Dimension-Aware Embeddings

Our target is to learn a dimension-aware embedding space where the pair-wise embedding distance is able to describe the distance between corresponding 3D dimensions so that the decoding process can be simplified. Before considering the consistency of distances across two spaces, we first define the distance criterion on the embedding space and the dimension space. Following the common practices in metric learning [56, 57], the Euclidean distance $\mathcal{D}(\mathbf{y}_i, \mathbf{y}_j)$ in (1) is used to measure the distance between embeddings.

$$\mathcal{D}(\mathbf{y}_i, \mathbf{y}_j) = \|\mathbf{y}_i - \mathbf{y}_j\|_2. \tag{1}$$

We further propose the variance-normalized Euclidean distance $\mathcal{J}(\mathbf{d}_i, \mathbf{d}_j)$ as the defined distance criterion on the target dimension space. Assume the dimension vector $\mathbf{d}_i = (h_i, w_i, l_i)$ and the standard deviations in the dataset

for height, width, and length are $(\sigma_h, \sigma_w, \sigma_l)$, the variance-normalized dimension distance is defined as (2):

$$\mathcal{J}(\mathbf{d}_i, \mathbf{d}_j) = \sqrt{\left(\frac{h_i - h_j}{\sigma_h}\right)^2 + \left(\frac{w_i - w_j}{\sigma_w}\right)^2 + \left(\frac{l_i - l_j}{\sigma_l}\right)^2}. \quad (2)$$

Considering different scales of the three elements (h, w, l) in the 3D dimension, the proposed distance function can equally measure the shape difference with respect to each element so that the distance will not be dominated by the element with larger scale. The empirical experiment in Sec. 4.4 demonstrates the variance-normalized distance can reflect the dimension difference better compared to the baseline Euclidean distance.

After the distance functions are defined on the embedding and dimension spaces, we then impose a constraint to encourage the inter-object distances on the embedding space to reflect their corresponding dimension distances. Motivated by recent progress in metric learning approaches [14, 58] for continuous labels, we utilize the dense log-ratio loss introduced in [58] to construct the dimension-aware embedding space. As shown in (3), the dense log-ratio loss samples all quadruplets (i, j, k, l) from the N object embeddings and minimizes the difference between the log-scale distance ratios from two different pairs (i, j) and (k, l). In other words, the embedding loss $\mathcal{L}_{embed}$ encourages the distance between any two embeddings to be proportional to the difference between their corresponding dimensions.

$$\mathcal{L}_{embed} = \frac{1}{2} \sum_{\{i,j\} \neq \{k,l\}} \left(\log \frac{\mathcal{D}(\mathbf{y}_i, \mathbf{y}_j)}{\mathcal{J}(\mathbf{d}_i, \mathbf{d}_j)} - \log \frac{\mathcal{D}(\mathbf{y}_k, \mathbf{y}_l)}{\mathcal{J}(\mathbf{d}_k, \mathbf{d}_l)} \right)^2. \quad (3)$$

In practice, we use the equivalent and more efficient formulation in (4) to compute the embedding loss $\mathcal{L}_{embed}$, as suggested by [58].

$$\mathcal{L}_{embed} = \frac{N(N-1)}{2} \sum_{i<j<N} \left(\log \frac{\mathcal{D}(\mathbf{y}_i, \mathbf{y}_j)}{\mathcal{J}(\mathbf{d}_i, \mathbf{d}_j)} \right)^2 - \left(\sum_{i<j<N} \log \frac{\mathcal{D}(\mathbf{y}_i, \mathbf{y}_j)}{\mathcal{J}(\mathbf{d}_i, \mathbf{d}_j)} \right)^2. \quad (4)$$

Since the utilized embedding loss imposes a strict constraint and the Euclidean distance is adopted to measure the discrepancy among embeddings, an underlying requisition is that the predicted embeddings should only contain dimension-specific information in order to satisfy the requirement. To this end, we employ an auxiliary dimension decoder $\mathcal{G}_{aux}$, which contains only one linear layer, to take the learned embeddings and predict the corresponding dimensions. The overall process is formulated as in (5).

$$\mathcal{L}_{embed\text{-}dim} = \left\| \mathbf{d} - \mathcal{G}_{aux}(\mathbf{y}) \right\|_1. \quad (5)$$

3.5. Shape Templates

Following the intuition that the underlying subcategories with different shapes can serve as useful information for dimension estimation, we propose to learn M shape templates and take advantage of the attention mechanism from transformers [46] to realize the information interaction between objects and templates.

Formally, we denote the dimension embeddings of objects as $\mathbf{Y} \in \mathbb{R}^{N \times C}$, where N is the number of objects and C is the embedding channel. The learnable shape templates are denoted as $\mathbf{T} \in \mathbb{R}^{M \times (C+3)}$, where M is the number of templates and the templates are further split into the template embeddings $\mathbf{T}_e \in \mathbb{R}^{M \times C}$ and the template dimensions $\mathbf{T}_d \in \mathbb{R}^{M \times 3}$. The attention module first uses linear layers to project the object embeddings $\mathbf{Y}$ to queries $\mathbf{Q}$ and the template embeddings $\mathbf{T}_e$ to keys $\mathbf{K}$ and values $\mathbf{V}$ as in (6),

$$\mathbf{Q} = \mathbf{Y}\mathbf{W}^q, \mathbf{K} = \mathbf{T}_e\mathbf{W}^k, \mathbf{V} = \mathbf{T}_e\mathbf{W}^v, \quad (6)$$

where $\mathbf{W}^q, \mathbf{W}^k, \mathbf{W}^v$ are the weight matrices of linear layers. The attention scores $\mathbf{A} \in \mathbb{R}^{N \times M}$ can then be computed with the scaled dot products between $\mathbf{Q}$ and $\mathbf{K}$, after which the attentive embeddings $\mathbf{T}_A$ from templates can be aggregated for object queries:

$$\mathbf{A} = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{D_k}}\right), \quad \mathbf{T}_A = \mathbf{A}\mathbf{V}, \quad (7)$$

where D_k is the number of channels in $\mathbf{Q}$ and $\mathbf{K}$. Since the attention scores reflect the pair-wise similarities between the objects and templates, we then compute the attention-guided combinations of M template dimensions as the initial dimension estimation $\tilde{\mathbf{D}}$ as in (8).

$$\tilde{\mathbf{D}} = \mathbf{A}\mathbf{T}_d. \quad (8)$$

To further refine the coarse dimensions $\tilde{\mathbf{D}}$, we combine the dimension embeddings $\mathbf{Y}$, the attentive embeddings from templates $\mathbf{T}_A$, and the initial dimension estimation $\tilde{\mathbf{D}}$ and process the fused features with linear layers to predict the residual dimensions, which are added to the coarse dimensions for output. The process is mathematically formulated as:

$$\hat{\mathbf{D}} = \text{cat}(\mathbf{Y} + \mathbf{T}_A, \tilde{\mathbf{D}})\mathbf{W}^d + \tilde{\mathbf{D}}, \quad (9)$$

where the coarse dimension estimation $\tilde{\mathbf{D}}$ is detached for stable gradients. We supervise both the predicted coarse and refined dimensions with L1 loss to fully utilize the power of residual learning:

$$\mathcal{L}_{temp\text{-}dim} = \|\tilde{\mathbf{D}} - \mathbf{D}\|_1, \quad \mathcal{L}_{dim} = \|\hat{\mathbf{D}} - \mathbf{D}\|_1. \quad (10)$$

Though the supervision is only applied to the predicted dimensions, the embeddings and dimensions of learnable templates can also be end-to-end updated through the aggregation process of the attention mechanism [46]. However,

Method	Extra data	3D@IoU=0.7			BEV@IoU=0.7			Runtime (ms)
		Easy	Mod.	Hard	Easy	Mod.	Hard	
MonoPSR [16]	LiDAR	10.76	7.25	5.85	18.33	12.58	9.91	120
AM3D [30]	Depth	16.50	10.74	9.52	25.03	17.32	14.91	400
PatchNet [29]	Depth	15.68	11.12	10.17	22.97	16.86	14.97	400
D4LCN [8]	Depth	16.65	11.72	9.51	22.51	16.02	12.55	200
Kinematic [2]	Temporal	19.07	12.72	9.17	26.69	17.52	13.10	120
MonoRUn [5]	LiDAR	19.65	12.30	10.58	27.94	17.34	15.24	70
CaDNN [38]	LiDAR	19.17	13.41	11.46	27.94	18.91	17.19	-
DD3D [34]	Depth	23.22	16.34	14.20	30.98	22.56	20.03	-
M3D-RPN [1]	None	14.76	9.71	7.42	21.02	13.67	10.23	160
SMOKE [26]	None	14.03	9.76	7.84	20.83	14.49	12.75	30
MonoPair [52]	None	13.04	9.99	8.65	19.28	14.83	12.89	57
RTM3D [21]	None	14.41	10.34	8.77	19.17	14.20	11.99	55
RAR-Net [24]	None	16.37	11.01	9.52	22.45	15.02	12.93	-
GrooMeD-NMS [17]	None	18.10	12.32	9.65	26.19	18.27	14.05	120
Ground-Aware [25]	None	21.65	13.25	9.91	29.81	17.98	13.08	50
MonoEF [60]	None	21.29	13.87	11.71	29.03	19.70	17.26	30
MonoFlex [55]	None	19.94	13.89	12.07	28.23	19.75	16.89	35
GUPNet [28]	None	20.11	14.20	11.77	-	-	-	-
AutoShape [27]	None	22.47	14.17	11.36	30.66	20.08	15.95	50
Monodle [31]	None	17.23	12.26	10.29	24.79	18.89	16.00	25.0
Monodle [31] + Ours	None	20.25	14.14	12.42	28.85	20.59	17.72	27.9
Improvement	-	*+3.02*	*+1.88*	*+2.13*	*+4.06*	*+1.70*	*+1.72*	-
GUPNet* [28]	None	21.80	14.25	11.72	29.90	19.52	17.24	28.6
GUPNet [28] + Ours	None	**23.62**	**16.10**	**13.41**	**32.82**	**21.98**	**18.70**	31.1
Improvement	-	*+1.82*	*+1.85*	*+1.69*	*+2.92*	*+2.46*	*+1.46*	-

Table 1. **The results of 3D/BEV object detection on KITTI *test* set.** The "Extra data" lists the required extra information. Within each group, the methods are sorted considering their performance on the moderate level of AP_{3D}. We highlight the best results without extra information in **bold**. The symbol "$*$" refers to our reproduced results with the officially released code.

we empirically find the attention scores across different templates tend to be very similar and the aggregation degenerates into simple averaging, which greatly reduces the actual capacity of the aggregated features. To alleviate this problem, we propose a simple regularization term $\mathcal{L}_{regu}$ to encourage sharpness in the attention scores:

$$\mathcal{L}_{regu} = -\sum_{i=1}^{N} \log\left(\max \mathbf{A}_i\right), \tag{11}$$

where $\mathbf{A}_i$ refers to the i_{th} row of the attention scores and represents the attentive similarities between the i_{th} object and M shape templates.

4. Experiments

4.1. Datasets

We evaluate the proposed method on the KITTI [10] bird's eye view and 3D object detection benchmarks, which include 7481 images for training and 7518 images for testing. We follow [6] to split the training images into *train* (3712) and *val* (3769) sets. We conduct all ablation experiments on the defined split and also report the test set results evaluated by the official server of KITTI. The average precision with 40 recall points $AP|_{R_{40}}$ is utilized as the main metric for evaluation. The detection results are evaluated on three levels of difficulty including easy, moderate, and hard. We mainly focus on the detection for the Car category with an IoU threshold of 0.7 and 0.5.

4.2. Implementation Details

We use the standard DLA-34 [54] as the backbone network, which extracts the 4x downsampled feature map with 64 channels. Every detection branch attached to the backbone consists of one $3 \times 3 \times 256$ conv layer, ReLU, and another $1 \times 1 \times c_o$ conv layer, where c_o is the output channel. The input image is padded to the size of 384 × 1280 for batch training. We set the embedding dimensionality as 256 and the number of shape templates as 4. The template embeddings are randomly initialized and the template dimensions are initialized with clustered dimensions from

Method	3D@IoU=0.7			BEV@IoU=0.7			3D@IoU=0.5			BEV@IoU=0.5		
	Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard
Monodle [31]	17.13	13.96	12.00	23.97	19.23	16.70	54.14	41.78	37.80	59.78	45.84	41.60
Monodle + Ours	20.82	15.64	13.82	28.21	21.67	18.82	58.94	45.06	39.56	63.75	48.68	44.24
AutoShape [27]	20.02	14.30	11.70	28.89	21.11	17.72	62.70	45.76	39.09	67.66	49.74	42.71
AutoShape + Ours	20.95	14.99	12.33	29.52	21.70	18.33	60.88	45.57	39.10	67.65	51.43	43.14
GUPNet [28]	20.53	14.70	12.77	28.53	20.97	17.77	59.94	43.93	39.37	66.47	47.85	43.17
GUPNet + Ours	23.47	16.19	14.15	31.51	22.84	19.55	59.67	44.06	39.42	66.17	49.44	43.17

Table 2. **The effectiveness of dimension embeddings based on different methods.** To demonstrate the generalization ability of the proposed dimension embeddings, we replace the dimension regression head in three state-of-the-art methods with our dimension embedding module and observe consistent improvements.

K-means [15]. We set the weights for $\mathcal{L}_{embed}$, $\mathcal{L}_{embed\text{-}dim}$, $\mathcal{L}_{temp\text{-}dim}$, $\mathcal{L}_{dim}$, and $\mathcal{L}_{regu}$ as 2, 1, 1, 1, and 0.05 and follow the settings from the baseline [31] for other losses. We train our model on two RTX 3090 GPUs for 140 epochs with the batch size as 16. We use the Adam optimizer with an initial learning rate of 0.00125 and weight decay as 10^{-5}. The warm-up strategy is used for the first five epochs and the learning rate is divided by 10 at 90 and 120 epochs. The augmentation of random cropping/scaling is adopted for 2D detection and dimension estimation and horizontal flipping is adopted for all downstream tasks.

When applying the proposed method to GUPNet [28] and Autoshape [27], we replace their dimension regression head with the proposed dimension embedding module and maintain other settings. The only difference is that we train GUPNet with the batch size of 24 on two GPUs instead of the batch size of 32 on 3 GPUs.

4.3. Main Results

Comparison with the state of the arts. To compare with existing state-of-the-art methods, we apply the proposed dimension embedding on the state-of-the-art methods Monodle [31] and GUPNet [28] and report the performance on KITTI *test* set. As shown in Table 1, the proposed dimension embedding module can significantly improve the performance of Monodle [31] and GUPNet [28], especially for hard objects. For example, the proposed module obtains a remarkable boost of 2.1 AP_{3D} over Monodle [31] under the hard setting. Second, we can outperform all existing methods without extra information when combining the stronger GUPNet [28] with the dimension embedding module. Compared with DD3D [34] which requires a large-scale depth dataset for pre-training, our method can achieve comparable performance and have no need for extra information. At the time of submission, our proposed method ranks 1st among all published methods for monocular 3D object detection on KITTI 3D object detection benchmark.

Latency analysis. To evaluate the extra latency induced overhead, we compute the average runtime with a batch size of 1 on a single RTX 2080Ti GPU. As shown in Table 1, the additional time cost from the proposed dimension embedding module is around 3ms for both Monodle [31] and GUPNet [28], which makes them still capable of real-time inference. Compared with the current best-performing method AutoShape [27], our method with GUPNet [28] baseline can achieve considerable improvement with only around 60% of the time for inference. Therefore, we demonstrate that the proposed module is both effective and lightweight and achieves an excellent trade-off between performance and efficiency.

	Easy	Mod.	Hard
baseline	23.97 / 17.13	19.23 / 13.96	16.70 / 12.00
+ embed.	24.02 / 17.23	19.77 / 14.26	17.01 / 12.14
+ embed. + std.	**26.12 / 18.22**	**21.05 / 14.78**	**18.24 / 13.06**

Table 3. **Ablation study on learning the dimension-aware embedding.** "embed." denotes using the dense log-ratio loss to learn the inter-object relations with embedding distances. "std." denotes using the variance-normalized distance to measure the similarities of dimensions. The effectiveness of learning dimension-aware embeddings is demonstrated.

Generalization analysis. Since most existing methods for monocular 3D object detection usually employ a simple regression head to estimate the dimension, our proposed method exactly focuses on the overlooked problem and is therefore orthogonal to these methods. To demonstrate the generalization ability of our dimension embeddings, we apply the proposed module to three recent state-of-the-art methods [27, 28, 31] and summarize the relative performance boost in Table 2. We can observe that applying the proposed dimension embedding module can consistently improve the performance of BEV/3D object detection, especially for the IoU threshold of 0.7.

4.4. Ablation Study

We conduct the ablation study on KITTI *val* set and utilize AP_{40} of the Car category for BEV/3D detection to quantitatively indicate the effectiveness of proposed modules.

Figure 4. **Qualitative results on KITTI *val* set.** We visualize the detection results of 3D object detection from GUPNet [28] equipped with our dimension embedding module. The detected cars, pedestrians, and cyclists are represented with cyan, light pink, and red bounding boxes. (Best viewed in color.)

Learning dimension-aware embeddings. As shown in Table 3, directly imposing the log-ratio loss can only bring negligible improvement. This is possible because the dimension distances are dominated by the larger lengths and the embeddings cannot obtain useful information when fitting the unbalanced distances. Therefore, the performance is observably improved with the variance-normalized distances among dimensions. These experiments demonstrate utilizing inter-object similarities can help to learn more discriminative embeddings.

Learning shape templates. We validate the design choices of shape templates based on the constructed dimension-aware embedding space. From Table 4, one can find the aggregation of features and shapes from learnable templates improves the performance significantly, only after the auxiliary dimension decoder is applied to supervise the dimension embeddings. Then, using the combined embeddings and coarse dimension predictions to perform refinement can further improve the performance. Finally, imposing the regularization term to encourage sharpness in attention scores can help to learn more representative templates and bring extra improvements.

4.5. Qualitative Results

We visualize the predicted 3D bounding boxes in Figure 4. One can observe that the proposed model can generate precise bounding boxes for objects at moderate distances. Meanwhile, our model also struggles with the clustered pedestrians, which can be considered a common problem for keypoints-based detectors.

5. Conclusion and Discussion

In this paper, we first point out that dimension estimation can serve as a key factor in recovering 3D locations and constructing 3D boxes. In order to improve the dimension estimation, we propose to leverage the inter-object similarities and the underlying subcategories as extra information. The former is considered by learning a dimension-aware embedding space, where the Euclidean distance can reflect the corresponding distance of dimensions. The latter is modeled with learnable shape templates, which are updated through the attention mechanism with the dimension embeddings of all objects. Extensive experiments on KITTI dataset demonstrate the proposed dimension embedding is effective, efficient, and can generalize to multiple methods.

	Easy	Mod.	Hard
embed. & std.	26.12 / 18.22	21.05 / 14.78	18.24 / 13.06
+ temp.	25.26 / 17.56	19.66 / 14.37	17.76 / 12.24
+ temp. + aux.	27.20 / 19.57	20.59 / 14.82	18.44 / 12.52
+ temp. + aux. + res.	27.23 / 19.62	21.40 / 15.32	18.66 / 13.58
+ temp. + aux. + res. + regu.	**28.21 / 20.82**	**21.67 / 15.64**	**18.82 / 13.82**

Table 4. **Ablation study on learning the shape templates.** "temp." denotes using the learnable shape templates. "aux." denotes using the auxiliary dimension decoder to supervise the learned embeddings. "res." denotes using the residual refinement for dimension estimation. "regu." denotes using the regularization loss in (11)

.

Limitations: Though the intuition of leveraging inter-object similarities is general, it is still unclear whether the proposed method can work for anchor-based methods. The concern is raised because the log-ratio loss tends to collapse when multiple anchors have the same target dimensions. We leave it for future work.

6. Acknowledgment

This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFA0700802, in part by the National Natural Science Foundation of China under Grant 62125603 and Grant U1813218, in part by Beijing Academy of Artificial Intelligence (BAAI), and in part by a grant from the Institute for Guo Qiang, Tsinghua University.

References

[1] Garrick Brazil and Xiaoming Liu. M3d-rpn: Monocular 3d region proposal network for object detection. In *ICCV*, 2019. 1, 2, 6

[2] Garrick Brazil, Gerard Pons-Moll, Xiaoming Liu, and Bernt Schiele. Kinematic 3d object detection in monocular video. In *ECCV*, 2020. 2, 6

[3] Yingjie Cai, Buyu Li, Zeyu Jiao, Hongsheng Li, Xingyu Zeng, and Xiaogang Wang. Monocular 3d object detection with decoupled structured polygon estimation and height-guided depth estimation. In *AAAI*, 2020. 1

[4] Fatih Cakir, Kun He, Xide Xia, Brian Kulis, and Stan Sclaroff. Deep metric learning to rank. In *CVPR*, pages 1861–1870, 2019. 3

[5] Hansheng Chen, Yuyao Huang, Wei Tian, Zhong Gao, and Lu Xiong. Monorun: Monocular 3d object detection by reconstruction and uncertainty propagation. In *CVPR*, 2021. 6

[6] Xiaozhi Chen, Kaustav Kundu, Ziyu Zhang, Huimin Ma, Sanja Fidler, and Raquel Urtasun. Monocular 3d object detection for autonomous driving. In *CVPR*, 2016. 1, 2, 6

[7] Xiaozhi Chen, Kaustav Kundu, Yukun Zhu, Huimin Ma, Sanja Fidler, and Raquel Urtasun. 3d object proposals using stereo imagery for accurate object class detection. *IEEE TPAMI*, 2017. 1

[8] Mingyu Ding, Yuqi Huo, Hongwei Yi, Zhe Wang, Jianping Shi, Zhiwu Lu, and Ping Luo. Learning depth-guided convolutions for monocular 3d object detection. In *CVPR*, 2020. 1, 2, 6

[9] Thanh-Toan Do, Toan Tran, Ian Reid, Vijay Kumar, Tuan Hoang, and Gustavo Carneiro. A theoretically sound upper bound on the triplet loss for improving the efficiency of deep distance metric learning. In *CVPR*, pages 10404–10413, 2019. 3

[10] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012. 2, 6

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 2

[12] Sungyeon Kim, Dongwon Kim, Minsu Cho, and Suha Kwak. Proxy anchor loss for deep metric learning. In *CVPR*, pages 3238–3247, 2020. 3

[13] Sungyeon Kim, Minkyo Seo, Ivan Laptev, Minsu Cho, and Suha Kwak. Deep metric learning beyond binary supervision. In *CVPR*, pages 2288–2297, 2019. 3

[14] Sungyeon Kim, Minkyo Seo, Ivan Laptev, Minsu Cho, and Suha Kwak. Deep metric learning beyond binary supervision. In *CVPR*, 2019. 5

[15] K Krishna and M Narasimha Murty. Genetic k-means algorithm. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, pages 433–439, 1999. 7

[16] Jason Ku, Alex D Pon, and Steven L Waslander. Monocular 3d object detection leveraging accurate proposals and shape reconstruction. In *CVPR*, 2019. 1, 2, 6

[17] Abhinav Kumar, Garrick Brazil, and Xiaoming Liu. Groomed-nms: Grouped mathematically differentiable nms for monocular 3d object detection. In *CVPR*, 2021. 6

[18] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *CVPR*, 2019. 1, 2

[19] Buyu Li, Wanli Ouyang, Lu Sheng, Xingyu Zeng, and Xiaogang Wang. Gs3d: An efficient 3d object detection framework for autonomous driving. In *CVPR*, 2019. 2

[20] Peiliang Li, Xiaozhi Chen, and Shaojie Shen. Stereo r-cnn based 3d object detection for autonomous driving. In *CVPR*, 2019. 1

[21] Peixuan Li, Huaici Zhao, Pengfei Liu, and Feidao Cao. Rtm3d: Real-time monocular 3d detection from object keypoints for autonomous driving. In *ECCV*, 2020. 1, 2, 3, 6

[22] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. Focal loss for dense object detection. In *ICCV*, 2017. 2, 3

[23] Lijie Liu, Jiwen Lu, Chunjing Xu, Qi Tian, and Jie Zhou. Deep fitting degree scoring network for monocular 3d object detection. In *CVPR*, 2019. 1, 2

[24] Lijie Liu, Chufan Wu, Jiwen Lu, Lingxi Xie, Jie Zhou, and Qi Tian. Reinforced axial refinement network for monocular 3d object detection. In *ECCV*, 2020. 6

[25] Yuxuan Liu, Yuan Yixuan, and Ming Liu. Ground-aware monocular 3d object detection for autonomous driving. *IEEE Robotics and Automation Letters*, 2021. 6

[26] Zechen Liu, Zizhang Wu, and Roland Tóth. Smoke: Single-stage monocular 3d object detection via keypoint estimation. *arXiv preprint arXiv:2002.10111*, 2020. 1, 2, 6

[27] Zongdai Liu, Dingfu Zhou, Feixiang Lu, Jin Fang, and Liangjun Zhang. Autoshape: Real-time shape-aware monocular 3d object detection. In *ICCV*, 2021. 1, 2, 3, 6, 7

[28] Yan Lu, Xinzhu Ma, Lei Yang, Tianzhu Zhang, Yating Liu, Qi Chu, Junjie Yan, and Wanli Ouyang. Geometry uncertainty projection network for monocular 3d object detection. In *ICCV*, 2021. 1, 2, 3, 6, 7, 8

[29] Xinzhu Ma, Shinan Liu, Zhiyi Xia, Hongwen Zhang, Xingyu Zeng, and Wanli Ouyang. Rethinking pseudo-lidar representation. In *ECCV*, 2020. 2, 6

[30] Xinzhu Ma, Zhihui Wang, Haojie Li, Pengbo Zhang, Wanli Ouyang, and Xin Fan. Accurate monocular 3d object detection via color-embedded 3d reconstruction for autonomous driving. In *ICCV*, 2019. 2, 6

[31] Xinzhu Ma, Yinmin Zhang, Dan Xu, Dongzhan Zhou, Shuai Yi, Haojie Li, and Wanli Ouyang. Delving into localization errors for monocular 3d object detection. In *CVPR*, 2021. 1, 2, 3, 4, 6, 7

[32] Arsalan Mousavian, Dragomir Anguelov, John Flynn, and Jana Kosecka. 3d bounding box estimation using deep learning and geometry. In *CVPR*, 2017. 1, 2, 4

[33] Yair Movshovitz-Attias, Alexander Toshev, Thomas K. Leung, Sergey Ioffe, and Saurabh Singh. No fuss distance metric learning using proxies. In *ICCV*, pages 360–368, 2017. 3

[34] Dennis Park, Rares Ambrus, Vitor Guizilini, Jie Li, and Adrien Gaidon. Is pseudo-lidar needed for monocular 3d object detection? In *ICCV*, 2021. 6, 7

[35] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *CVPR*, 2018. 2

[36] Zengyi Qin, Jinglu Wang, and Yan Lu. Monogrnet: A geometric reasoning network for 3d object localization. In *AAAI*, 2019. 1, 2

[37] Zengyi Qin, Jinglu Wang, and Yan Lu. Triangulation learning network: from monocular to stereo 3d object detection. In *CVPR*, 2019. 1

[38] Cody Reading, Ali Harakeh, Julia Chae, and Steven L Waslander. Categorical depth distribution network for monocular 3d object detection. In *CVPR*, 2021. 6

[39] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, pages 815–823, 2015. 3

[40] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *CVPR*, 2020. 1

[41] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *CVPR*, 2019. 1, 2

[42] Andrea Simonelli, Samuel Rota Bulo, Lorenzo Porzi, Manuel López-Antequera, and Peter Kontschieder. Disentangling monocular 3d object detection. In *CVPR*, 2019. 2

[43] Andrea Simonelli, Samuel Rota Bulò, Lorenzo Porzi, Elisa Ricci, and Peter Kontschieder. Towards generalization across depth for monocular 3d object detection. In *ECCV*, 2020. 2

[44] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *CVPR*, pages 4004–4012, 2016. 3

[45] Jiaming Sun, Linghao Chen, Yiming Xie, Siyu Zhang, Qinhong Jiang, Xiaowei Zhou, and Hujun Bao. Disp r-cnn: Stereo 3d object detection via shape prior guided instance disparity estimation. In *CVPR*, 2020. 1

[46] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 2, 5

[47] Jiang Wang, Yang Song, Thomas Leung, Chuck Rosenberg, Jingbin Wang, James Philbin, Bo Chen, and Ying Wu. Learning fine-grained image similarity with deep ranking. In *CVPR*, pages 1386–1393, 2014. 3

[48] Jian Wang, Feng Zhou, Shilei Wen, Xiao Liu, and Yuanqing Lin. Deep metric learning with angular loss. In *ICCV*, pages 2593–2601, 2017. 3

[49] Xun Wang, Xintong Han, Weilin Huang, Dengke Dong, and Matthew R Scott. Multi-similarity loss with general pair weighting for deep metric learning. In *CVPR*, pages 5022–5030, 2019. 3

[50] Yan Wang, Wei-Lun Chao, Divyansh Garg, Bharath Hariharan, Mark Campbell, and Kilian Q Weinberger. Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving. In *CVPR*, 2019. 1, 2

[51] Zetong Yang, Yanan Sun, Shu Liu, and Jiaya Jia. 3dssd: Point-based 3d single stage object detector. In *CVPR*, 2020. 1

[52] Chen Yongjian, Tai Lei, Sun Kai, and Li Mingyang. Monopair: Monocular 3d object detection using pairwise spatial relationships. In *CVPR*, 2020. 1, 2, 3, 4, 6

[53] Baosheng Yu and Dacheng Tao. Deep metric learning with tuplet margin loss. In *ICCV*, pages 6490–6499, 2019. 3

[54] Fisher Yu, Dequan Wang, Evan Shelhamer, and Trevor Darrell. Deep layer aggregation. In *CVPR*, 2018. 2, 3, 6

[55] Yunpeng Zhang, Jiwen Lu, and Jie Zhou. Objects are different: Flexible monocular 3d object detection. In *CVPR*, 2021. 1, 2, 3, 6

[56] Wenzhao Zheng, Zhaodong Chen, Jiwen Lu, and Jie Zhou. Hardness-aware deep metric learning. In *CVPR*, pages 72–81, 2019. 4

[57] Wenzhao Zheng, Jiwen Lu, and Zhou Jie. Structural deep metric learning for room layout estimation. In *ECCV*, 2020. 3, 4

[58] Wenzhao Zheng, Jiwen Lu, and Jie Zhou. Structural deep metric learning for room layout estimation. In *ECCV*. Springer, 2020. 5

[59] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019. 2, 3

[60] Yunsong Zhou, Yuan He, Hongzi Zhu, Cheng Wang, Hongyang Li, and Qinhong Jiang. Monocular 3d object detection: An extrinsic parameter free approach. In *CVPR*, 2021. 6